UNIVERSIDAD AUTONÓMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Tumour associated microbiota: comparison of methods for the identification of low abundance microbiome samples using RNA sequencing data

Máster Universitario en Bioinformática y Biología Computacional

Autor: Francisco José Jurado Rueda
Tutora: Lola Alonso Guirado
Supervisora: Núria Malats Riera
Ponente: Miguel Redondo Nieto

Grupo Epidemiología Genética y Molecular. Centro Nacional de Investigaciones Oncológicas.

Febrero 2020

Acknowledgements

To my mother, for her tireless
support and constant supply
of tuppers.

To my bike, for the time saved.

To my entire group, for keeping
up a great environment in the lab.

To my directors Núria and Lola.

May I leave here a piece of
Lola's uniqueness:

Francisella gaditana
Bacteria del alma mía
No vas a estar en vejiga
Porque a ti te dé la gana

Yo no me creo tu género
Y mucho menos tu especie
¿Cómo vas a estar ahí dentro
escondida entre los genes?

Francisella gaditana
Yo te relego al olvido
O me aportas una prueba
O eres falso positivo

INDEX

ABSTRACT

We have conducted a benchmarking study with *in silico* tools for microorganism detection: PathSeq (GATK), Kraken2, MetaPhlAn2, and DRAC. The first three algorithms are publicly available, whereas the fourth is an in-house pipeline. Data used as input was non-human reads obtained from RNA sequencing of the gene expression of human bladder tumours from TCGA project databases. Each tool has its own database, against which aligns the input RNA sequences.

A previous process of preparation and adaptation had to be arranged since each tool has its own requirements, regarding files format, hardware, and software resources. We also dealt with other issues during the "set up" itself, such as running time. We addressed the speed issue by splitting the dataset into batches and parallelizing, when possible. We built up a comparable table format for each output. Since tool launched its own file format this drawback was overcome by parsing each one specifically. A customized BAM file was designed to calculate True Positive Rate and True Negative Rate.

A comparison of the absolute outcome was made along with α-diversity and file by file correlation analysis. Available metadata were used to look for clusters in the sample's distribution after a PCA with absolute abundances was performed. To seek for relationships with selected metadata, we applied generalized linear models to test potential associations. A gold standard/simulation dataset has been built by combining human and known bacteria reads in a common file. Precision/positive-predictive-value and recall/sensitivity indicators were estimated to assess the performance of each tool.

PathSeq was the slowest tool, taking almost 42.3 days to finish. Kraken was the best tool in terms of recall (0.69) while maintaining its precision as high as the rest. On the other end, DRAC obtained the lowest recall results (0.29), followed by MetaPhlAn very close. No significant association was found between any microorganism species and the two features explored ("gender" and "tumour stage").

Kraken was the fastest and most sensitive tool. DRAC is the least sensitive tool when considering all species tested, whereas when regarding only bacteria MetaPhlAn is the least sensitive. PathSeq is by far the slowest tool. Contaminant bacterial species were found within our samples.

Keywords

Microbiome, RNA sequencing, bladder cancer, Kraken, PathSeq, MetaPhlAn, BAM format, read counts, gold standard, negative control.

1. INTRODUCTION

Cancer is still one of the most commonly extended illnesses. Last year there were up to 18 million new cases diagnosed worldwide, with almost 300,000 detected in Spain. Bladder cancer (BC) is the tenth most common type (Bray F, 2018). This is a complex and multifactorial disease, and both genetic predisposition, environmental exposures, and lifestyle behaviour, could trigger its development.

It has been reported that 15-20% of cancer is linked to viral, parasitic or bacterial infections (Garrett, 2015). Some of these causal relationships are already known, for instance *Helicobacter pylori* associated with gastric cancer and Human Papilloma Virus (HPV) with cervical cancer, among others. The interaction host-microbes is specific and there are several paths through which they can influence oncogenesis, tumour progression, and response to anticancer therapy. These microbe-associated cancers can indirectly damage host DNA by increasing local inflammation and/or producing reactive oxygen species. They can do it directly, as well by integrating their own genome into the host's, in the case of a virus. In the case of bacteria, they can segregate genotoxins such as colibactin, and modify signalling pathways resulting in immunodepression (Gagnaire, 2017).

In the case of BC, it is known that *Schistosoma haematobium*, a flatworm, lays its eggs in the bladder muscle causing irritation and local inflammation that could eventually derive in cell proliferation and apoptosis reduction, both typical features of bladder carcinogenesis. This was one of the first infectious agents associated with BC. Therefore, its protumorigenic actions are well known. However, no bacteria taxa have been found to be causal for BC, this might be because, until recently, urine was thought to be sterile in healthy individuals (Robles C, 2013). This incorrect assumption is due to old culture-dependent technologies (Dematei, 2017).

Nowadays, and thanks to high throughput techniques such as Next Generation Sequencing (NGS) and to *in silico* tools for DNA mapping, we can explore the urine and the normal/tumour tissue microbiota. Regarding BC, Bucevic *et al.* identified an Operational Taxonomic Unit (OTU) differentially represented in a case control study of 11 healthy individuals *vs* 12 BC patients. The OTU belonged to *Fusobacterium* genus (Bucevic, 2018). Also, Bi *et al.* found a significantly different microbiome pattern between BC cases and controls. For instance, *Actinomyces europaeus* was specifically abundant among BC cases (Bi, 2019). Both contributions amplified 16S rDNA and used it to assess the microbiota present in tissue samples.

Although several efforts, as the ones mentioned above, have been published in microbiota of BC, no previous study has been conducted using transcriptomics data from these tumours to gather bacteria species using the non-human sequences. International consortia have focused in tumour gene expression,

disregarding the unmapped reads while we are actually taking advantage of them.

Currently scientific community has no optimal classification program/pipeline for detecting microorganisms from human transcriptomic data. Therefore, a benchmarking comparison is necessary in order to pinpoint the best tool in terms of performance, precision and sensitivity.

2. OBJECTIVES

The main goal of the study is to perform a benchmarking of four publicly available tools for microbe sequences recognition in RNA-Seq data: PathSeq (GATK), Kraken2, MetaPhlAn, and DRAC. To this end, we will test and compare their running time, power and resources required, as well as, the precision and recall performance estimates.

Secondary objectives are:

1. To explore the taxonomic results each tool yields by checking common and specific taxa groups.
2. To explore downstream ways to filter out possible false positives.
3. To explore potential associations between the microbiome profiles and patients' characteristics.

4. HYPOTHESES

Our working hypothesis were:

1. While each tool yields different results due to their own internal algorithm, similarities in results will be found on microbiome's abundances, richness, and α-diversity.

2. Kraken2 is the fastest algorithm on the basis of the *k*-mers procedure it applies.

3. MetaPhlAn2 is the least sensitive tool due to the unique "gene-markers" strategy it uses.

4. We expect to find certain differences in the presence of bacteria between tumour and peritumoral tissue samples. Particularly bacterial species typically related to BC such as *Fusobacterium nucleatum* and *Actinomyces europaeus* will be present in greater amount within tumour and not in peritumoral tissues.

5. We expect to find certain bacterial species typically related to operating contamination such as *Staphylococcus epidermidis* and *Propionibacterium acnes,* among others, in a large percentage of samples.

5. MATERIALS AND METHODS

The four *in silico* tools that were compared ahead were selected due to their recurrent usage in microbiome sequences recognition. Many of new tool released on this field, compares itself with the four of them selected to prove its competitiveness (Gihawi, 2019). They are the state-of-the-art examples when it comes to taxonomic sequences classification and taxonomic labels assignation.

4.1 CLASSIFIERS DESCRIPTION

I. **Kraken2**

Kraken was developed by Derrick Wood, Jennifer Lu, and Ben Langmead at the Johns Hopkins University in 2014.

Kraken2 is based on an exact '*k*-mer' alignment. It splits each query sequence into *n* pieces called *k*-mers (by default is 31 bases long). The number of *k*-mers depends on the read and *k*-mer length. Its database is also organised in the same structure. Each *k*-mer is associated with a taxon.

Kraken2 works with one *k*-mer at a time, assigning it to particular taxon. Once all the *k*-mers of a given sequence are assigned, the taxa are weighted according to the number of *k*-mers. The one with the highest rank is then assigned to a particular sequence. If two taxa have the same number of *k*-mers assigned, the final label of the sequence will be the Lowest Common Ancestor (LCA) found in the phylogenetic tree. (Wood, 2014).

II. **MetaPhlAn2**

MetaPhlAn was developed by Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower at The Huttenhower Lab, Department of Biostatistics, Harvard T.H. in 2012.

MetaPhlAn is based on marker genes. Its database is a reduced collection of characteristic genes preselected from coding sequences that unequivocally identify specific microbes. No previous read filtering is required (such as error detection, assembly or gene annotation) since spurious reads will very rarely match any marker gene.

It uses UCLUST to align the row sequences to the catalog of marker genes with a threshold of 75% of identity. This procedure is recursively applied from genera level to phyla taxa. (Nicola Segata, 2012)

III. PathSeq (GATK)

PathSeq was developed by Aleksandar D Kostic, Akinyemi I Ojesina, Chandrasekhar Pedamallu, Joonil Jung, Gad Getz, and Matthew Meyerson at the Broad Institute in 2012.

PathSeq is based on three consecutive filters. It discards low quality, low complexity, duplicated and human-related reads. For this purpose, it uses several filters such as DUST and Bloom filters. This step removes most of the human reads, since even the reads with a single human *k*-mer are discarded.

The second filter is carried out by BWA-MEM aligner which will use human genome as a reference to get rid of any remaining read that might have passed the previous filter. For the last filter it uses again BWA-MEM and queries different microbial genomes in order to classify taxonomically the non-human remaining reads. (Walker, 2018)

IV. DRAC

DRAC was developed by Lola Alonso at the Genetic and Molecular Epidemiology Group, Spanish National Cancer Research Centre (CNIO).

The name stands for Discarded Reads Alignment and Coverage. It takes a BAM file as input and dismiss the human-mapped sequences, then it maps the rest of the sequences against a reference database (bacterial genomes downloaded from RefSeq) using BWA. The unique aspect of this pipeline is that it considers the coverage along the bacterial genome, using Bedtools and evaluates the expected *vs* observed coverage.

Knowing how long they are and how much they should cover, DRAC computes an internal score to discard those reads with low coverage. By doing so, it discards bacterial assignments with very poor coverage which are probably (FP). (No published yet)

4.2. INPUT DATA

We used gene expression data from bladder cancer patients obtained from The Cancer Genome Atlas (TCGA) which is a public and open repository dependent of the National Cancer Institute in the USA. It contains detailed molecular information from over 20,000 primary cancer patients and matched normal samples spanning 33 cancer types. Most of the BC samples belonged to

408 patients with muscle invasive subtype, with a total of 433 BAM files. 21 of them were tumour-adjacent normal samples.

My starting point was the 433 BAM files already free of human reads. Since they were already aligned against the human genome, “samtools” was used to get rid of the host’s reads. By doing so, only the unidentified (unmapped) reads remained (N=1,167,115,490), and we discarded 64,005,943,937 reads. We saw also a clear reduction of the data weight, from 3.9 TB to 55 GB.

DRAC and PathSeq were directly fed with the BAM files whereas Kraken2 and MetaPhlAn only accepted fastq files. For that purpose, we used again “samtools” to convert BAM into fastq format as seen in Box 1. Although we fed MetaPhlAn with two fastq files corresponding to matching paired-end read, this tool did not consider it. It just concatenates all of the reads and disregards that “paired-end” information.

```
samtools fastq -1 fg_1.fq -2 fg_2.fq output.bam
```

Box 1: samtools command to convert fastq files into bam files.

The four programs were run in the same workstation with 32 cores and 96 GB RAM memory.

4.3. TOOL PERFORMANCE

To measure an algorithm’s performance, it is necessary to know, how well it classifies the sequences. This means to classify the sequences according to whether they are true positive (TP), false positive (FP), true negative (TN) and false negative (FN) and calculate the correspondent rates. To get these parameters, we needed an already known dataset with controlled numbers of reads coming from well sequenced species. No public, nor available website or paper supplementary dataset fulfills these requirements. Therefore, we built this simulation dataset and used it as our “gold standard”.

The main idea behind this procedure was to separate TP from FP. We did so by constructing two BAM files, the gold standard with human plus microbial reads combined, and the negative control, just with human reads. Results obtained from gold standard file were a mix of TP and FP. Microbial reads might be correctly labelled, or human reads might be classified as microbial as well. On the other hand, result obtained from negative control were only FP. By subtracting to the gold standard results, the negative control results we kept just the TP.

As negative control, we used the same human reads initially discarded. In this case, we kept them using the opposite command, as they were aligned against human genome in the BAM file, we saved the ones aligned. We sampled reads, file by file, until they summed up 1 million reads, this file was in BAM format. We did this so by gathering reads from all files. The command is shown in Box 2.

```
for bamfile in `ls /BLCA/blcaRNAseq/*/*.bam`;do samtools view -s 0.000017
-F 12 $bamfile; done > ~/humanTCGA/mixedbladder.sam
```

Box 2: samtools command to extract one million of only human reads from all 433 bamfiles. The parameter -s extracts this proportion of random reads from each file. The -F parameter excludes all the unaligned reads.

This gold standard contained controlled number of reads from selected genomes. We included 6 species coming from the top list of abundances in TCGA applying the four tools, plus 2 specific species that were detected only by each tool. All of them were run with very same conditions since they all worked with the same species.

For that purpose, we downloaded the complete genome of *Escherichia coli, Lactobacillus lactis, Propionibacterium acnes, Lactococcus raffinolactis, Staphylococcus epidermidis,* and *Rothia mucilaginosa*. These species corresponded to the positive controls, we also included as the positive control some reads from other species only were recognised by each tool. Species detected just by PathSeq were *White clover mosaic virus* and *Oryza sativa alphaendornavirus.* Species detected just by MetaPhlAn were *Dasheen mosaic virus* and *Bovine alphaherpesvirus.* DRAC and Kraken shared all their species with the other tools, that is why we could not select specific species from them.

For the gold standard BAM file, random sequences of 48 nucleotides were selected from the microbiome genomes mentioned, leaving a 80-120 nucleotide gap between them. This gap represents the insert not sequenced in the middle of two members of the same DNA fragment. However, for this second read, we computed the reverse complementary out of it in 50% of the cases.

We made sure to set the bitwise flag of each read according to its position and alignment status which means:

- First sequence: 77 “Read paired, Read unmapped, Mate unmapped, First in pair”
- Second sequence: 141 “Read paired, Read unmapped, Mate unmapped, Second in pair”

The rest of the 7 fields including “reference sequence name”, “1-base leftmost position”, “CIGAR”, “Reference of the mate”, “Position of the mate”, “Observed template length”, “Segment sequence” were properly set to either “0” or “*” as empty. This step was performed a thousand times for each genome, thus we eventually got ten thousand paired-end sequences in SAM format. The Python script used for extracting reads from genomes and building the gold standard is available in Appendix I.

Negative control file previously generated, was duplicated and combined with the microbial reads, using samtools to that purpose. The reads shuffled performed was also done by samtools. As shown in Box 3.

```
#Merge: merged.bam will be the output
samtools merge merged.bam mixedbladder.bam generated.sam
#Shuffle
samtools collate -Ou merged.bam shuffled > shuffled.bam
```

Box 3: samtools commands to first merge and then shuffle the reads.

Across the original 433 BAM files, there was 1% of non-human reads on average. Initially, we stack to that proportion for our gold standard dataset, meaning 10,000 species reads, because that was the average proportion of non-host reads found among TCAG files. We increased the proportion up to 10%, 100,000 microbial species reads, to explore the role of abundance in the performance (resolution) and in computational needs of the tools.

Regarding the quality field of the BAM file, we did 2 rounds: One with low quality per base ('#' character in the ASCII 32 code) and a second with high quality per base ('A' character in the ASCII 32 code). Note that we will only show results for quality reads in 10% abundance for PathSeq. This is due to an issue we encountered during the tool set-up combined with a shortage of time. The flag: '-disable-tool-default-read-filters', which is set "false" by default, literally performs a prefiltering of our reads. This is quite inconvenient, and we figured it out just in time for the last trial (quality reads in 10% abundance).

4.4. TIME AND PARALLELIZATION

Taking into account the volume of data we were dealing with, we first tried to parallelize the inner processes of the tools. Time and computational power would be saved by doing so. It was then when we encountered a fact, that we named "PathSeq paradox". PathSeq was faster when it was not run in parallel mode. A surprising fact without a clear explanation yet, we even let the developers team know about it.

We also split the 433 files names into 9 batches and launched 9 processes simultaneously.

4.5. COMMON FILE STRUCTURE

Each tool launched its own output file according to its internal design. Each one with a different structure as shown in Figure 1.This fact let us to come up with a solution since the variety of format output files, was not appropriate for comparison purposes.

The desirable scaffold for the final matrix would be 433 columns, one for each initial BAM file, and as many rows as bacteria discovered for the richest file. We built it for each tool and we came up with the resulting tables (Appendix II).

There was a problem with MetaPhlAn output because it did not include TaxID number for the discovered bacteria, by contrast it just considered their name. The

TaxID or taxonomic identifier, is an international accession number that can be queried with the NCBI taxonomy resource. The lack of TaxID for some of the program was a caveat for comparison purposes since it could lead to error. For instance, Kraken and MetaPhlAn recognised the same species *Cutibacterium acne*s but Kraken uses the new genus name for this species, labelling it like *Cutibacterium acnes*. To overcome this issue, we used the NCBI's Taxonomy (Federhen, 2012) browser as it labelled each species name with its taxID number. For the ones that it couldn't recognise, we curated them manually.

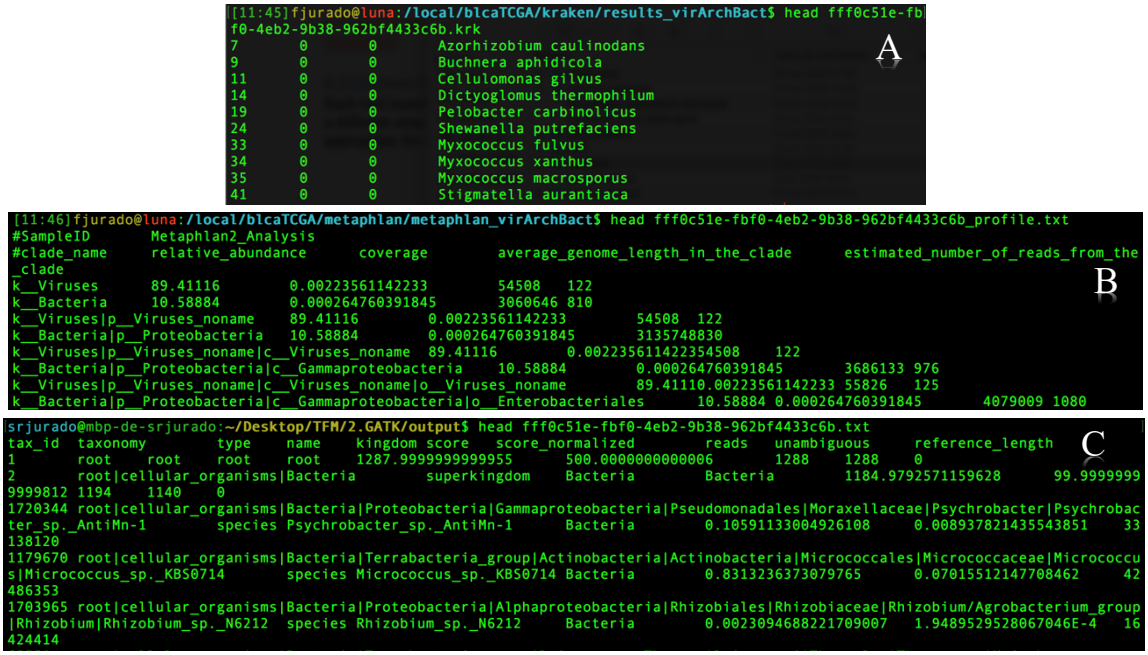


Figure 1: kraken output file (A) contains 4 columns and 12,409 rows; MetaPhlAn output file (B) contains 5 columns and 24 rows; PathSeq output file (C) contains 9 columns and 6,290 rows.

4.6. METADATA ASSOCIATION

Metadata information was available for the 433 files corresponding to 408 patients. We downloaded them from the Genome Data Commons Portal repository maintained by the National Cancer Institute. This table contained up to 79 features distributed in columns.

We wondered whether there was an association between the species pattern for each file obtained with the information regarding those files and the patients. For that end we used the function `glm()` belonging to the package {stats} in R. `glm()` is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution.

We focused just in "gender" and "tumour stage" features. The model was fed with the normalized abundances, otherwise we couldn't compare files, since the raw counts are not scaled to the total counts. To calculate the normalized abundance of a given species, the number of reads assigned to it was divided by the total

number of reads within the file. We ended up having the normalized abundance of each species in this file. We then computed the Odds Ratio and adjusted the *p*-value and the false discovery rate based on Benjamini-Hochberg method.

4.7. TOOLS SETUP

Each tool has its own flags and parameters. We tried to adjust them as much as possible by setting either on or off the flags or tuning up the parameters we considered more appropriated for each of the tool trials.

Notice that DRAC is an in-house pipeline which uses BWA as aligner for the non-human reads and samtools for setting up the filters. Finally, it computes the coverage of the reads using "genomeCoverageBed" command. Is not a public tool yet, therefore there is not a simple nor short command to launch it. Therefore, we will not report on DRAC's setup here.

Box 4 shows the command used to initiate MetaPhlAn.

```
metaphlan2.py fg_1.fq,fg_2.fq --bowtie2out gs.bowtie2.bz2
-t rel_ab_w_read_stats --nproc 6 --input_type multifastq > profiled_gs.txt &
```

Box 4: command used to launch MetaPhlAn.

Kraken allows you to set a "confidence" threshold as shown in Box 5. This score was especially useful when not all of a read's *k*-mer aligned against the same reference (genome). A read was labelled as a certain species only if the average number of *k*-mers to this species was greater than the confidence score.

```
kraken2 --threads 16 --db refSeqBAVkraken –output readbasis_ow.txt
--report-zero-counts --fastq-input $inputfq $inputfq2 --gzip-compressed
--paired --confidence 0.90 --report $prefixinput.rep
```

Box 5: command used to launch Kraken.

For example, we were able to solve the "misclassified" 45 conserved nucleotides between *Pseudomonas tolaasii* and phage Phi X 174. Before tuning this parameter, we found many (N=14,000) reads classified as *Pseudomonas tolaasii* coming from a single file. We assembled these reads using SPAdes (Nurk S, 2017) with *k*-mers length 7,15, and 21 as shown in Box 6. Out of this assembly trial, no scaffold was obtained, moreover, the longest contig formed had 69 nucleotides long. This fact suggested that probably *P. tolaasii* was not really present in the sample.

```
spades.py -1 p_t_forward.fa -2   p_t_reverse.fa -t 1 -m 6 -k 7,15,21
-- only-assembler -o tolaasii.out
```

Box 6: command used to launch spades.

Phi X 174 is used as an internal control for the sequencing process in Illumina. The origin of these reads was much more plausible to be Phi X 174 than *P. tolaasii.* The "confidence" threshold helped us getting rid of false positive cases such as this one.

Box 7 shows the command used to initiate PathSeq.

```
gatk PathSeqPipelineSpark --input $bamfile --filter-bwa-image
pathseq_host.fa.img --kmer-file pathseq_host.bfi --min-clipped-read-length
48 --microbe-fasta pathseq_microbe.fa --microbe-bwa-image
pathseq_microbe.fa.img --taxonomy-file pathseq_taxonomy.db --output
&nombre.bam --scores-output $nombre.txt 2> $nombre.log
```

Box 7: command used to launch PathSeq.

6. RESULTS

The files that were analysed had all the same structure: 433 columns and as many rows as species they found in the richest sample. Also, files were normalised depending on the comparison performed.

5.1 TIME AND PARALLELIZATION

PathSeq eventually required 32,930,991 seconds (381 days) to complete the analysis. When we considered the 9 time-overlapping batches, the running reduced to 42.3 days. The time required when processing the biggest file (1.4 GB of non-human reads) is shown in Table 1.

File/Tools	PathSeq	Kraken2	Metaphlan2
2775d5a7-9663-4169-83e3-937032a27d78	6d/10h/56m/42s	5m/2s	15m/6s
1.4GB (18,353,550 reads)			

Table 1: Process time required for each tool to process the biggest file.

5.2 MICROORGANISMS TAXA FOUND

We set 'species', within the taxonomic chain, as the level for comparison for this benchmarking because of the well-known taxa. Despite the four tools were obviously fed with the same 433 files, their outcomes were non-identical, mainly due to the databases consulted by each of them. The results differed basically in three aspects: abundance, correlation and biodiversity.

5.2.1 ABUNDANCE

There were 23 common species shared all cross the three tools. Kraken discovered 2,509 species and shared 23 with MetaPhlAn along and combined with PathSeq. MetaPhlAn discovered 155, the lowest, and shared 33 PathSeq. PathSeq recognised up to 13,122 species and shared with Kraken 1,609.

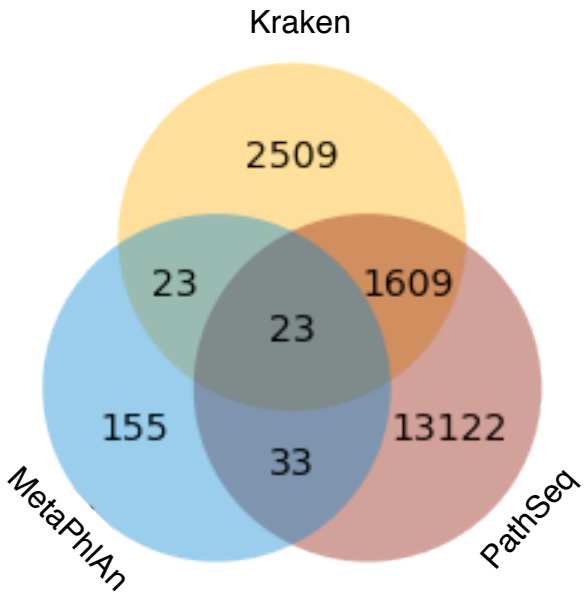


Figure 2: Venn diagram of the species overlapping across the three tools.

We observed in Figure 3 a clear agreement in the file by file absolute number of reads, shared across the 4 tools, DRAC included.

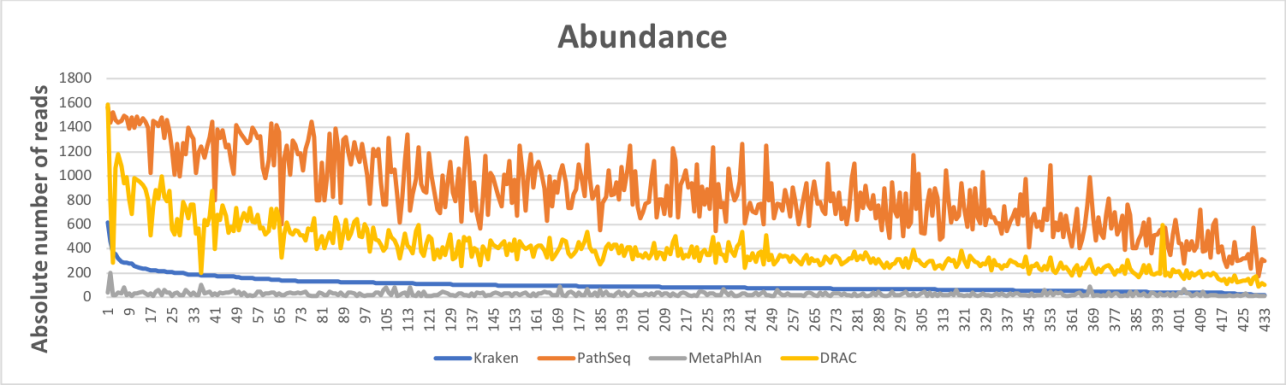


Figure 3: Sample by sample species abundance correlation. Samples were sorted according to the abundance present in Kraken's output files.

However, the absolute number of reads assigned to each species depended on each tool. For instance, we could see in Table 2 that *Rothia mucilaginosa* is among the top 6 species detected by Kraken but not by MetaPhlAn. Note that the top 3 bacteria species were consensually maintained for the three tools *Cutibacterium acnes*, *Lactococcus lactis and Escherichia coli*.

Taxonomic ID	Species name	Kraken	Gatk	MetaPhlAn
1747	Cutibacterium acnes	127112	171645	128655
1358	Lactococcus lactis	92034	176217	549575
562	Escherichia coli	16992	217612	53526
1282	Staphylococcus epidermidis	6801	28581	1151
1308	Streptococcus thermophilus	4893	21405	5379
43675	Rothia mucilaginosa	4816	21193	678
1229751	Lactococcus phage BM13	3598	5563	19659
40324	Stenotrophomonas maltophilia	2798	9542	795
40215	Acinetobacter junii	2210	34551	2436
40214	Acinetobacter johnsonii	1403	57578	1538
1262537	Lactococcus phage P680	909	5496	10215
1366	Lactococcus raffinolactis	529	66307	35968
274	Thermus thermophilus	475	1909	1148
1262535	Lactococcus phage jm2	436	4677	9161
729	Haemophilus parainfluenzae	395	2980	684
1262538	Lactococcus phage phi7	316	4115	161
1304	Streptococcus salivarius	221	11171	95
2047	Rothia dentocariosa	76	15050	356
12239	Pepper mild mottle virus	66	295	305
114416	Lactococcus phage ul36	63	1974	230
12321	Alfalfa mosaic virus	16	205	288
294369	Solenopsis invicta virus 1	2	6	5
12227	Tobacco etch virus	1	1	2
447604	Tomato yellow leaf curl Vietnam betasatellite	0	0	4
11867	Avian myelocytomatosis virus	0	0	5
656025	Gossypium darwinii symptomless alphasatellite	0	0	4
39720	Walleye dermal sarcoma virus	0	0	7

Table 2: First 27 species, sorted by kraken´s results. There are 23 consensuses.

As we can see in Table 3, when adding DRAC to the comparison, the number of species dropped from 23 to 7 consensus species. This suggests that DRAC restricts the number of species because it uses only bacterial genomes and not viruses. Notice that top bacteria remains almost in the same position.

tax_id	Species_name	Kraken	Gatk	MetaPhlAn	DRAC
1747	Cutibacterium acnes	127112	171645	128655	284108
562	Escherichia coli	16992	217612	53526	655182
1282	Staphylococcus epidermidis	6801	28581	1151	47732
1308	Streptococcus thermophilus	4893	21405	5379	23684
40215	Acinetobacter junii	2210	34551	2436	26764
1366	Lactococcus raffinolactis	529	66307	35968	24620
1304	Streptococcus salivarius	221	11171	95	7308

Table 3: Total 7 species shared with all 4 tools. Sorted by Kraken's results.

5.2.2 CORRELATION

Having a previous knowledge of the regular microbiota of the bladder, might help giving credit to possible TP and dismiss possible FP. Still, there is no real way to determine what species could be FP. These tools are not perfect, they all have an error margin for misclassifying a certain read and also Illumina reads are error-prone. Nonetheless if two of them come up with the same label from a

particular read, the possibility of error decreases. This possibility would be even lower if the three of them agree.

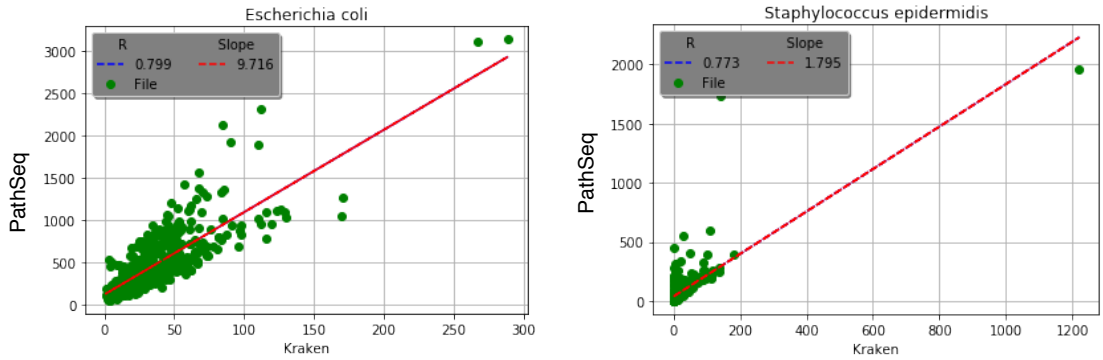


Figure 4: Scatterplot of Kraken's and PathSeq's results where correlation values are shown. On the left side, the case of *Escherichia coli*. On the right side, the case of *Staphylococcus epidermidis*.

That is why we decided to plot for each microorganism, the number of reads in each file given by one tool against other tool, as seen in Figure 4. This is a way to show how reliable is the presence of a species. The closer the correlation coefficient to 1, the more similar are the results and thus, the more credible. The slope refers to the "agreement" proportion between tools.

The correlation raised among tools when comparing species typically related to contamination, such as *E. coli* showed in Figure 4 and others like *P. acnes.* Probably because this is due to the fact that the presence of these bacteria is real.

5.2.3 BIODIVERSITY AND RICHNESS

Biodiversity of metagenomics samples can be stablished using several measures: Richness, α-diversity, and β-diversity. Richness is the number of different species found within the sample. Second α-diversity is the variety of OTUs in a defined habitat and Shannon index is a way to calculate it (Wooley JC, 2010). Lastly, β-diversity provides a measure of the degree to which samples differ from one another regarding their composition. (Goodrich JK, 2014)

Before any comparison was done among results from the tools, we normalised the outcomes to the total number of non-human reads per file. Approximately 1% of all the reads within the 433 BAM files were not aligned to the human reference genome, *ergo*, there were not of human origin. That was the number used for standardization.

We used the Shannon index to measure α-biodiversity. We checked Simpson Index as well and results were very similar. We used the package `vegan` in R for computing both indexes. As shown in Figure 5, there was significant differences among the three tools. PathSeq yielded the highest diversity of species: 2 points greater than the already consider "upper limit" of the index (being 5 the "upper limit"). MetaPhlAn was on the other side of the spectrum, Shannon index value 1. Shannon index values calculated using Kraken profiles were between 2 and 3.

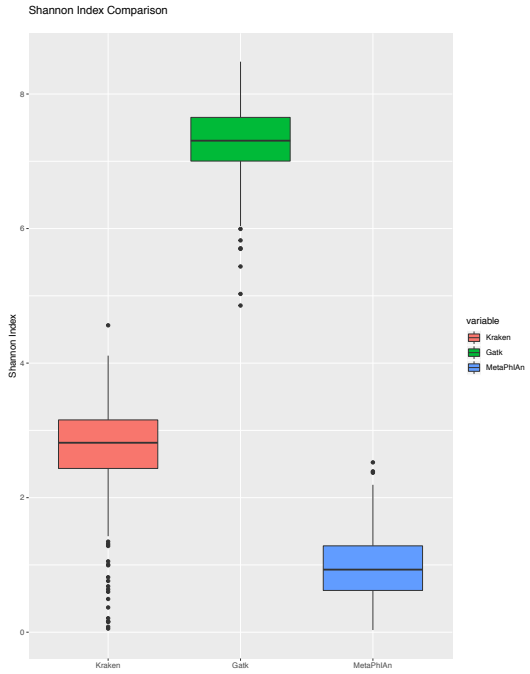


Figure 5: Boxplot of the Shannon index of the 433 files according to three tools

When we checked the correlation among files, there was a clear and common decreasing pattern shared by all tools when sorting by Kraken richness. We observe this phenomenon in Figure 6.

On the other hand, that correlation was very low in terms of biodiversity. PathSeq and MetaPhlAn were constant when sorting by Kraken. We can observe this behaviour in Figure 7.

On average, tools did not yield the same α-diversity (Figure 5). Certainly, they failed to agree on every sample's α-diversity as shown in Figure 7.

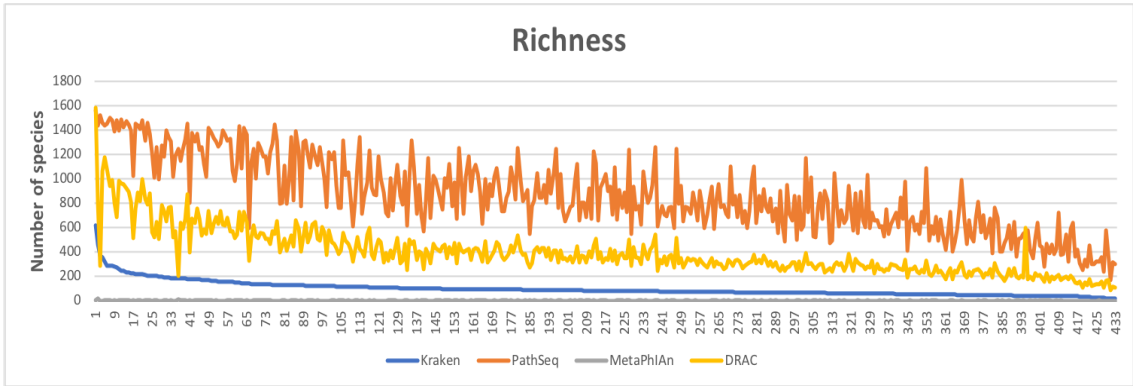


Figure 6: Sample by sample richness correlation. Samples were sorted according to the number of species present in Kraken's output files.

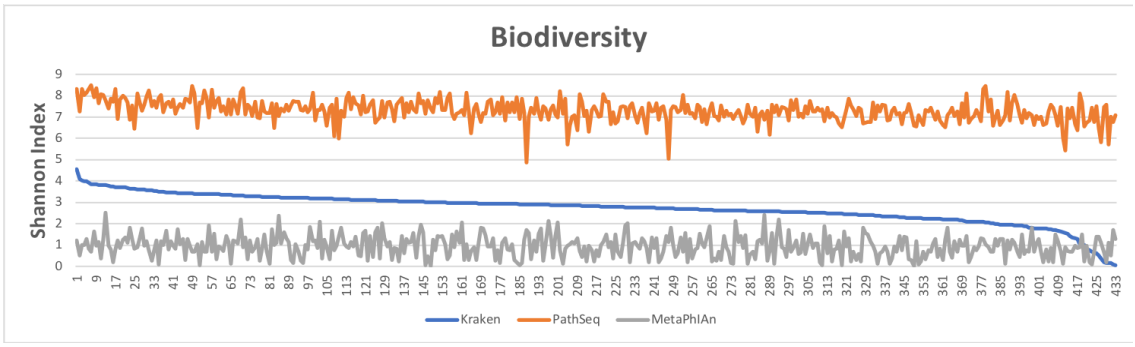


Figure 7: Sample by sample biodiversity correlation. Samples were sorted according to Shannon index of Kraken's files.

5.3 TUMOUR VS NORMAL TISSUE COMPARISON

We started with 433 BAM files from 408 patients, which means that some subjects were sampled more than once. These 21 samples whose patient ID is duplicated, came from solid adjacent tissue, resected from bladder but without tumoral compartment.

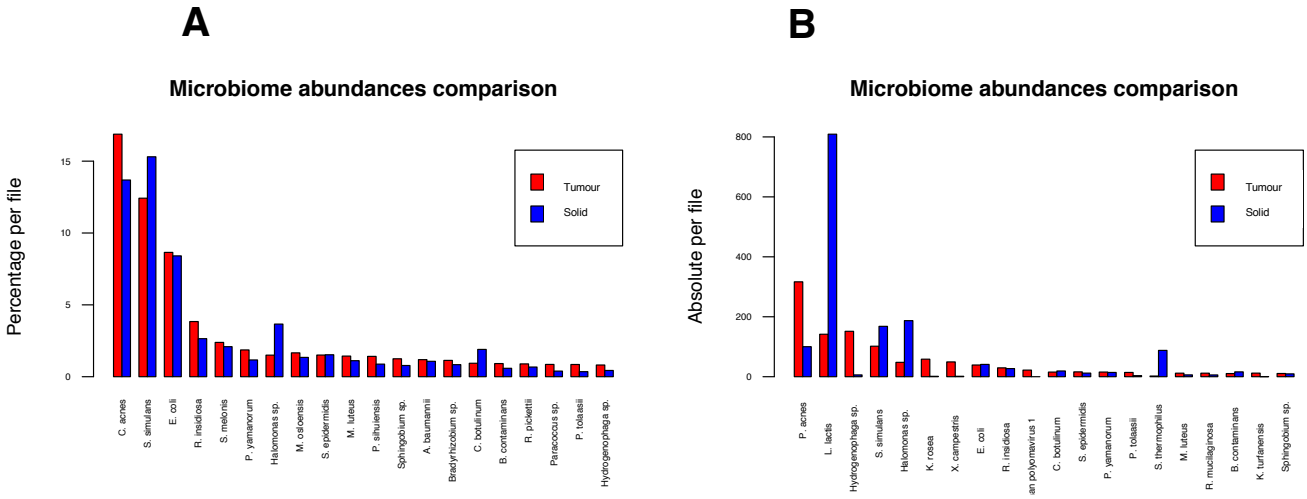


Figure 8: Top 20 species comparison, showing tumour *vs* normal tissue values. A: relative abundance averaged among files. B: absolute abundance averaged among files. Note that *L. lactis* is not in the top 20 species when representing the relative abundance.

The 20 most abundant species were selected to perform a comparison between tumour and solid adjacent tissue. As shown in Figure 8. Regarding the relative abundance, there was no clear difference among them, not even when checking standard deviation. In 12 out of 20 species, the percentage of reads per file was greater in tumour than in solid. In the case of absolute abundance, in the case of relative abundance it was prominent the amount of *L. lactis* reads among normal tissue samples. However, standard deviations do widely overlap, meaning there is no a big difference.

As mentioned previously, there were several bacteria species that seem to be associated with BC*: F. nucleatum* (Bucevic, 2018) and *A. europaeus* (Bi, 2019). Thus, we checked whether these two species were present in the output of any of the tools tested.

DRAC and MetaPhlAn did not recognise any of them, not in the tumour samples nor in the adjacent ones. Kraken found in 21 of the tumour samples, *F. nucleatum* reads. On average, within those samples, relative abundance of *F. nucleatum* was 0.19%. Kraken classified no reads as *A. europaeus.*

PathSeq yielded that *F. nucleatum* was present in 181 of tumour samples and 8 of solid samples. On average, within those tumour and solid adjacent samples, relative abundance of *F. nucleatum* was $4.23 \cdot 10^{-5}$ % and $4.85 \cdot 10^{-5}$ % respectively.

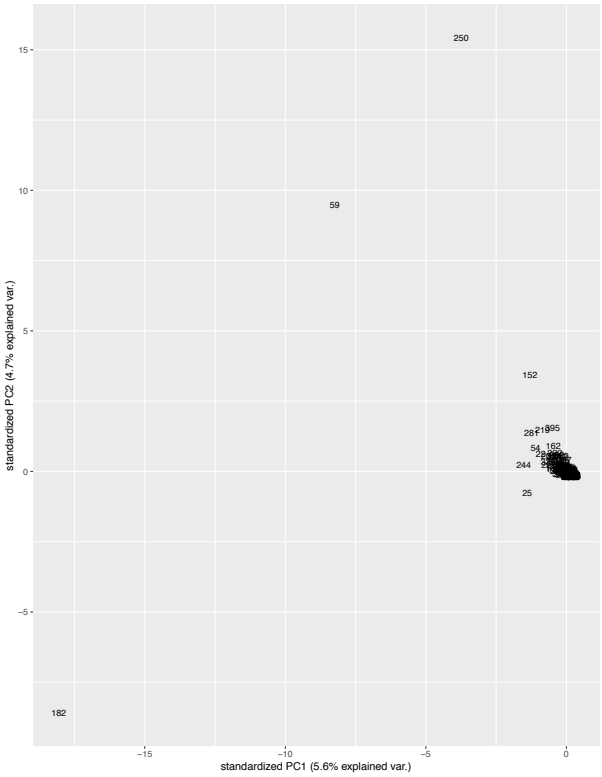
According to PathSeq, *A. europaeus* was present in 246 of tumour and 11 of solid adjacent samples. On average, within those tumour and solid adjacent samples, relative abundance of *A. europaeus* was $4.46 \cdot 10^{-5}$ % and $4.64 \cdot 10^{-5}$ % respectively.

5.4 PRINCIPAL COMPONENT ANALYSIS

Files with similar characteristics would, ideally, fall together and would be observed as a cluster. For each sample we had a similar number and disposition of microorganism's assigned reads. Each dimension referred to a specific microorganism's abundance. The size of the dimension was tool dependent, since it could be as large as 30,057 for PathSeq, 12,408 for Kraken, 279 for MetaPhlAn or 2,453 for DRAC.

For the sake of visualization, we first removed the absent microorganisms and secondly performed a PCA to the remaining features (species). By doing these two steps in advance, we saved time and computational power. We used for that purpose R function `prcomp()` of the package "stats". For each tool we plotted the top two principal components, the ones that better explain the variance in our data.

5.4.1 KRAKEN



PCA was done with the remaining species after removing zeros. Remaining taxa were 2,509.

We observed that in general, PCA could not find explanatory components that explain the variance of our data (Figure 9):
PC1 just explain 5.6% of the total variance. Most of the dots seem to be quite similar, remaining all of them tightly together.
No clear cluster nor significant distribution appeared to exist, nonetheless we plotted according to available metadata.

Figure 9: PCA plot coming from Kraken's raw counts.

There was no clear clustering when labelling by any of the 79 available features. In Figure 10 we show just two of them, gender, and tumour stage.

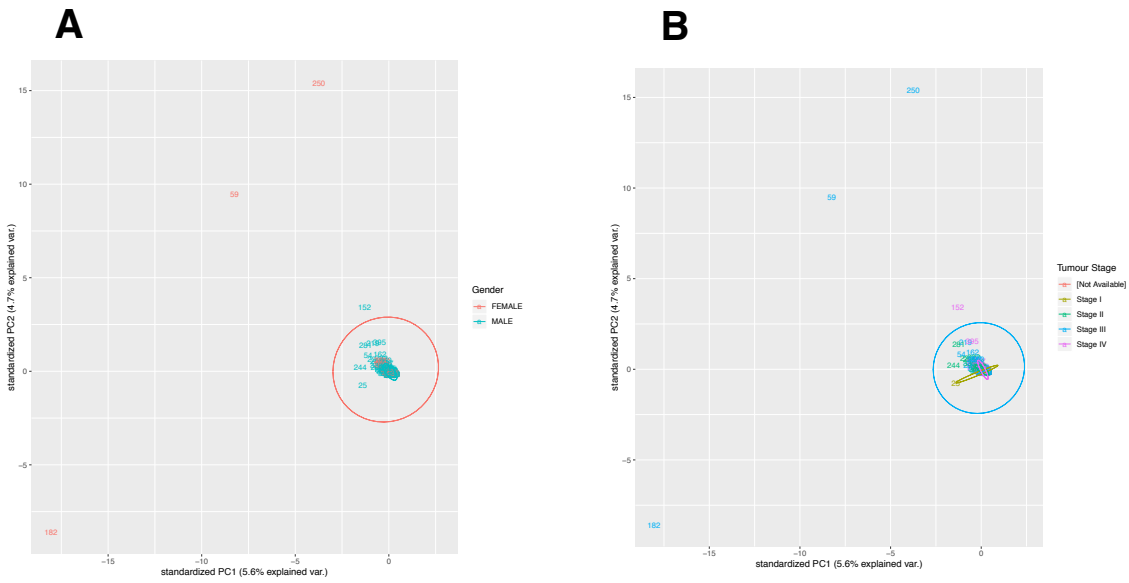


Figure 10: PCA plot coming from Kraken's output. A: coloured by gender. B: coloured by tumour stage.

5.4.2 PATHSEQ

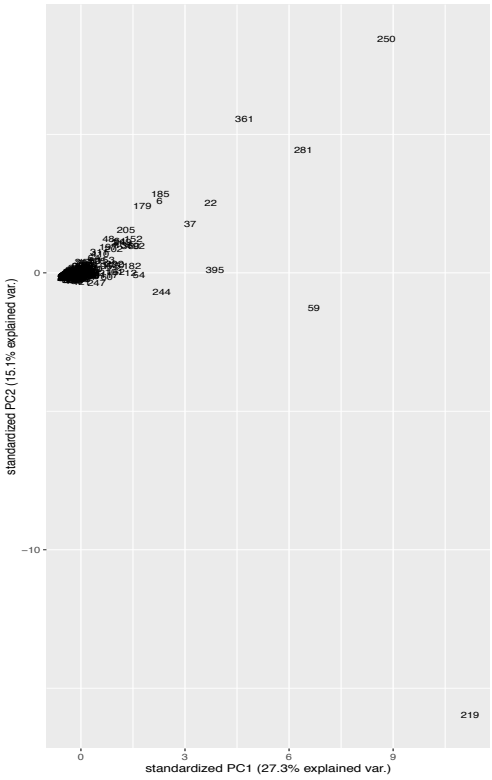


Figure 11: PCA plot coming from PathSeq's output.

PCA was done with the remaining species after removing absent ones. Remaining taxa were 13,281.

The results of the PCA were more suitable for plotting since the variance explained was wider and we observed a horizontal distribution. In this case, the first component explained almost a third of the variance of our files (Figure 11).
There was no clear clustering when labelling by any of the 79 available features. We show just two of them, gender and the metastasis status as shown in Figure 12.

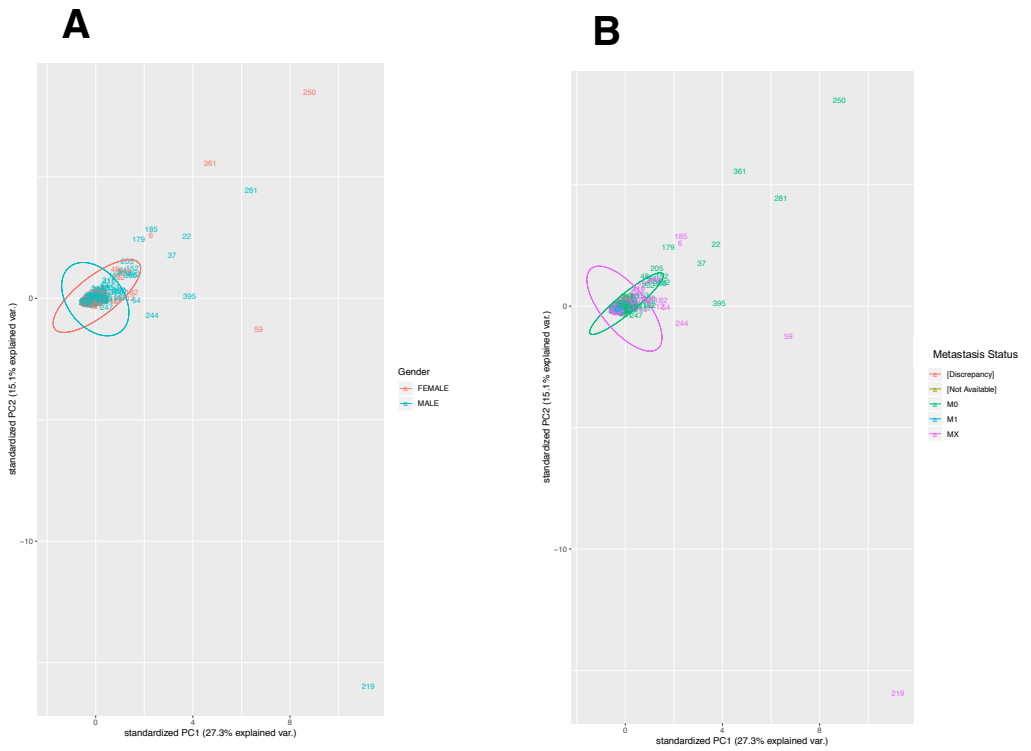


Figure 12: PCA plot coming from PathSeq's output. A: coloured by gender; B: coloured by metastasis status.

We also performed a biplot of the PCA results seeking the bacteria responsible of this apparent horizontal distribution (first dimension of the data). We kept the 45 first species when sorting by abundance. Surprisingly the variance explained was 50%.

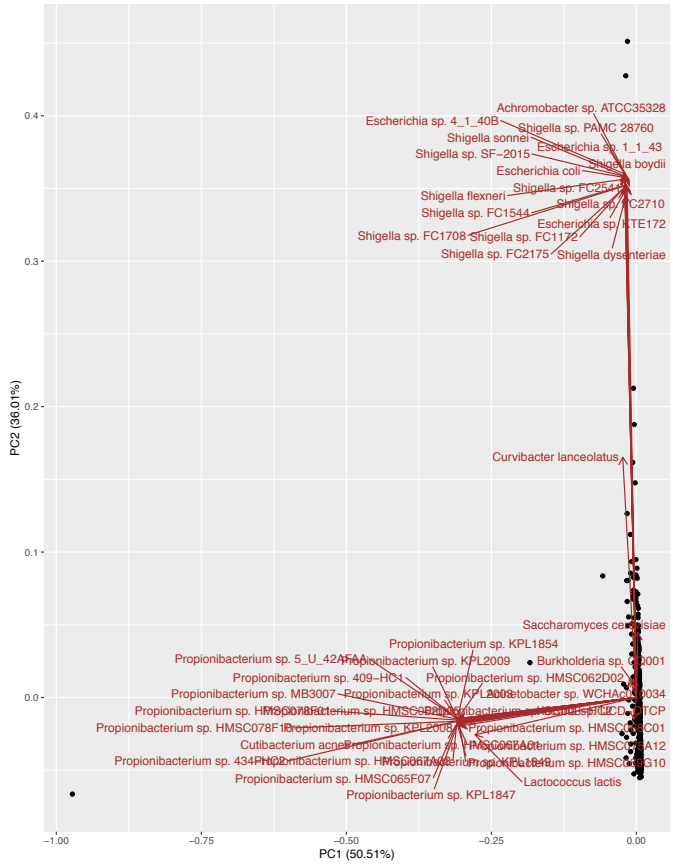


Figure 13: Biplot of top 45 most abundant species from PathSeq's outcome.

We observed that the whole distribution had two main axes (Figure 13).

We also observed two well separated clusters. The first one mostly formed by *Propionibacterium* genus, the main reason of the horizontal distribution. The second cluster, formed mainly by *Escherichia* and *Shigella* genera, stretched the dots distribution upwards.

Probably the outlier placed in the bottom-left corner explained by itself much of the horizontal variance.

5.4.3 METAPHLAN

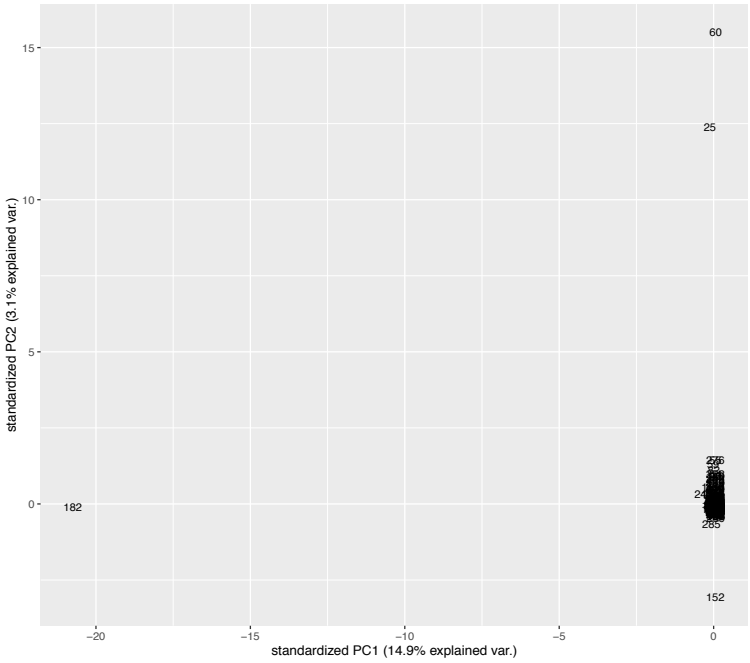


Figure 14: Plot of the PCA results from MetaPhlAn's output.

PCA was done with the remaining species after removing zeros. Remaining taxa were 158.
In this case, PCA could not find new dimensions to explain the variance within our data.

Note that the best principal component explained just the 15% of the variance due to the outlier sample 182.

The second principal component covered a 3% of the variance (Figure 14).

5.5 METADATA ASSOCIATION

5.5.1 KRAKEN

The species significantly associated with gender and tumour stage (Table 4). None of the associations held significance after correcting for false discovery rate with Benjamini-Hochberg method. Odds Ratio (OR) is used to represent this association.

Gender

	Species Name	OR	pvalue	padj
1290	Staphylococcus hominis	7.199757e−01	0.02623060	0.9969068
225991	Comamonas aquatica	8.030575e−01	0.04509246	0.9969068
2219696	Sphingomonas sp. FARSPH	3.330338e+00	0.03687380	0.9969068
1612173	Niveispirillum cyanobacteriorum	2.289303e−05	0.02711609	0.9969068
106648	Acinetobacter bereziniae	4.085919e−01	0.04728514	0.9969068
294	Pseudomonas fluorescens	6.755725e+00	0.02431418	0.9969068

Tumour Stage

	Species Name	OR	pvalue	padj
152682	Sphingomonas melonis	8.621107e+02	0.042975839	0.9900866
178899	Carboxydocella thermautotrophica	6.267222e+13	0.047134537	0.9900866
444444	Chelatococcus daeguensis	4.914667e+78	0.034082817	0.9900866
146827	Corynebacterium simulans	6.376430e−56	0.041269444	0.9900866
28449	Neisseria subflava	3.113902e−67	0.033102362	0.9900866
1479019	Methylobacterium sp. C1	1.538932e−146	0.003198402	0.9900866

Table 4: List of species most associated with gender (left side) and tumour stage (right side), although none of them were significantly associated after the BH correction.

Figure 15 displays the distribution of the relative abundance percentage for all the files when clustering by the feature of study. The ones plotted are the ones

with the minimum *p-value*. The case of *Sphingomonas melonis* in the tumour stage has a clearer difference between classes due to the larger OR.

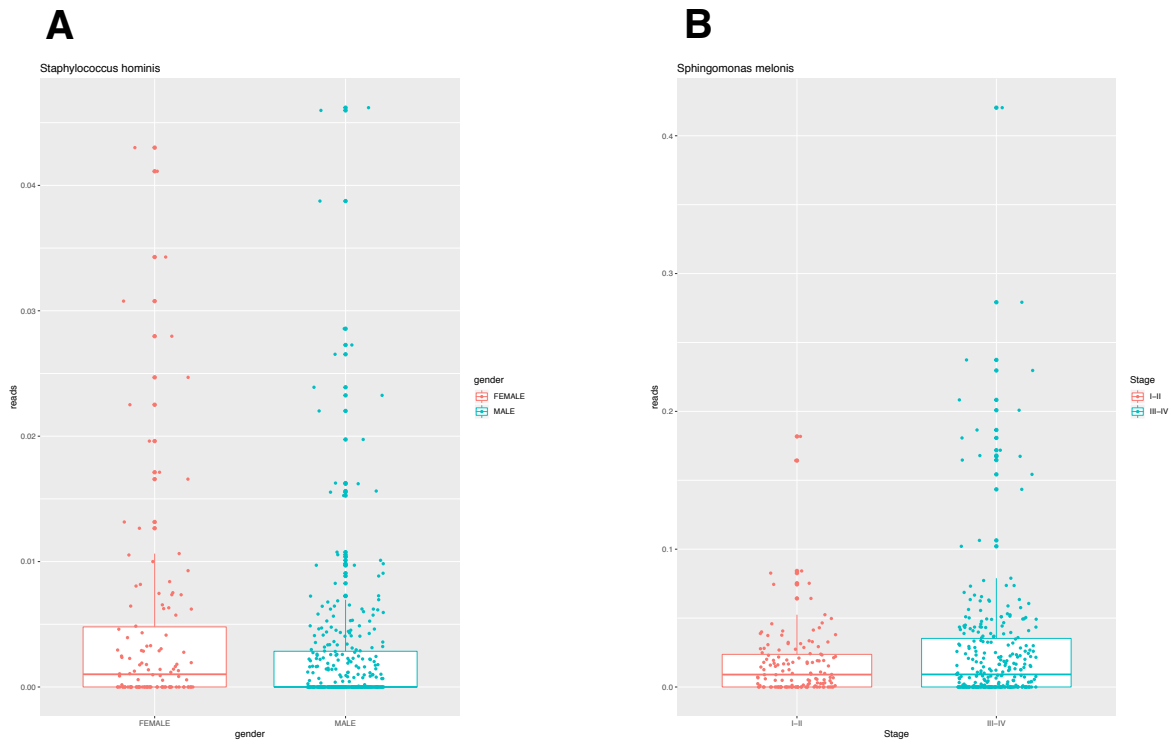


Figure 15: Relative abundance boxplot of two bacteria not significantly associated with gender (A) and tumour stage (B) according to Kraken results

5.5.2 PATHSEQ

The species significantly associated with gender and tumour stage are shown in Table 5. None of the associations held significance after correcting for false discovery rate with Benjamini Hochberg method.

Gender

	Species Name	OR	pvalue	padj
36651	Penicillium digitatum	0.24769968	0.0004438402	0.4682733
746128	Aspergillus fumigatus	0.04619988	0.0034885429	0.9988601
36630	Aspergillus fischeri	0.04173085	0.0042428932	0.9988601
5062	Aspergillus oryzae	0.03530176	0.0017272302	0.7329093
5061	Aspergillus niger	0.06112962	0.0148820030	0.9988601
393283	Pestalotiopsis fici	0.04773297	0.0138830040	0.9988601

Tumour Stage

	Species Name	OR	pvalue	padj
1922217	Candidatus Erwinia haradaeae	1.159333e−101	0.0000434420	0.5769531
545275	Thioalkalivibrio sp. ALE20	6.969635e−97	0.0003127334	0.7229448
357794	Psychromonas ingrahamii	3.005628e−26	0.0004028034	0.7229448
314282	Psychromonas sp. CNPT3	3.431395e−25	0.0005233800	0.7229448
1461322	Vibrio renipiscarius	1.768925e−42	0.0007486598	0.7229448
126385	Providencia alcalifaciens	3.035913e−33	0.0008606887	0.7229448

Table 5: List of species most associated with gender (left side) and tumour stage (right side), although none of them were significantly associated after the BH correction.

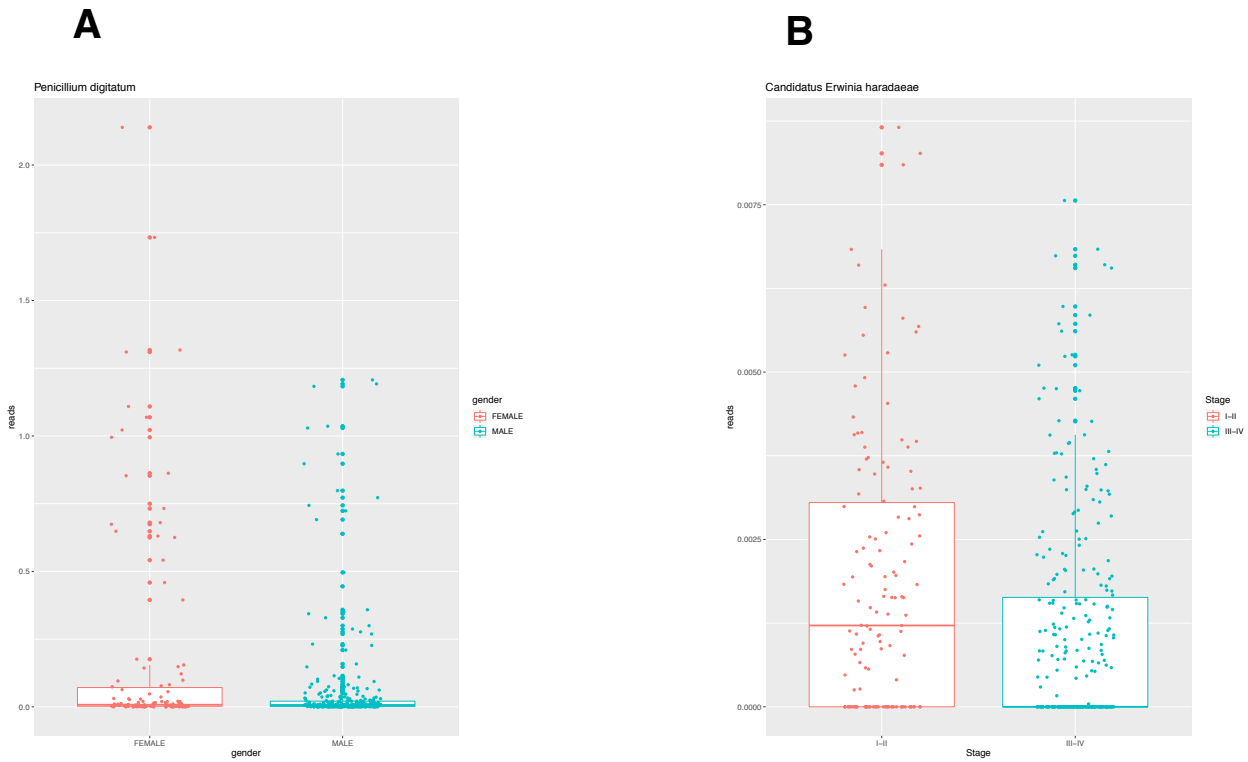


Figure 16: Relative abundance boxplot of two bacteria not significantly associated with gender (A) and tumour stage (B) according to PathSeq results.

Figure 16 displays the distribution of the relative abundance percentage for all the files when clustering by the feature of study. The ones plotted are the ones with the minimum *p-value*.

5.5.3 METAPHLAN

Interestingly, *Curvibacter lanceolatus* is the one with the smallest *p-value* for both features (Table 6), as shown before.

Gender

	Species Name	OR	pvalue	padj
86182	Curvibacter lanceolatus	9.675585e−01	0.1919969	0.9883969
4932	Saccharomyces cerevisiae	9.907110e−01	0.2125468	0.9883969
11867	Avian myelocytomatosis virus	5.860499e−01	0.3269349	0.9883969
871699	Turdivirus 1	9.164163e−02	0.3529239	0.9883969
362693	Oryza sativa endornavirus	9.241387e−119	0.3576278	0.9883969
130310	Human adenovirus D	3.033522e−01	0.3578131	0.9883969

Tumour Stage

	Species Name	OR	pvalue	padj
86182	Curvibacter lanceolatus	9.438390e−01	0.07132255	0.9844965
871699	Turdivirus 1	1.255825e−15	0.11451330	0.9844965
1367671	Malvastrum leaf curl Philippines betasatellite	2.320802e−01	0.16635477	0.9844965
1747	Propionibacterium acnes	9.673529e−01	0.17190167	0.9844965
509923	Beet cryptic virus 1	8.244127e−01	0.19163782	0.9844965
11885	Fujinami sarcoma virus	4.833138e−01	0.21136496	0.9844965

Table 6: List of species most associated with gender (left side) and tumour stage (right side), although none of them were significantly associated after the BH correction.

Unfortunately, the *p-value* is not low enough to consider significant the association between every bacterium and the corresponding variable.

5.5.4 DRAC

Finally, DRAC yielded microorganisms not significantly associated with gender nor with tumour stage as listed in Table 7.

No plot of the file-by-file relative abundance percentage is shown in this case since no significant association has been found.

Gender

	Species Name	OR	pvalue	padj
237609	Pseudomonas alkylphenolica	4.844524e+05	0.0006195016	0.9744481
399741	Serratia proteamaculans 568	7.802096e−04	0.0019806168	0.9744481
656178	Pandoraea vervacti	1.373650e−12	0.0055859941	0.9744481
1427342	Pseudomonas aeruginosa SCV20265	6.081788e−01	0.0071766667	0.9744481
301	Pseudomonas oleovorans	4.632976e−01	0.0114638681	0.9744481
56956	Thermus brockianus	1.404994e−08	0.0154552799	0.9744481

Tumour Stage

	Species Name	OR	pvalue	padj
42239	Streptomyces sampsonii	4.311034e−08	0.003653258	0.9490414
1300165	Kosakonia cowanii JCM 10956 = DSM 18146	1.896041e−04	0.003801434	0.9490414
1813821	Shigella sp. PAMC 28760	5.860835e+01	0.005860254	0.9490414
104087	Pseudomonas frederiksbergensis	7.392974e−02	0.005906653	0.9490414
1357916	Sphingopyxis sp. QXT−31	1.164575e−02	0.006627507	0.9490414
465817	Erwinia tasmaniensis Et1/99	1.902688e−04	0.007403611	0.9490414

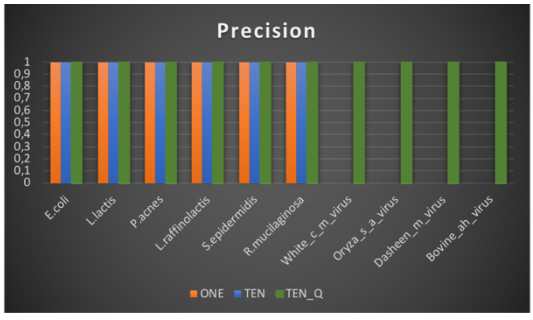
Table 7: List of species most associated with gender (left side) and tumour stage (right side), although none of them were significantly associated after the BH correction.

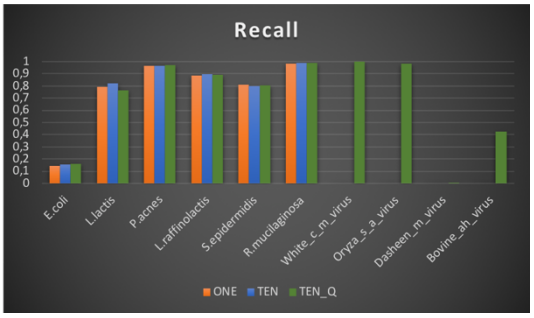
5.6 GOLD STANDARD

Below we show the comparison of the performance of each tool in three scenarios: 1) one percent microbial reads and bad quality base calls (orange); 2) ten percent microbial reads and bad quality base calls (blue); 3) ten percent microbial reads and good quality base calls (green).

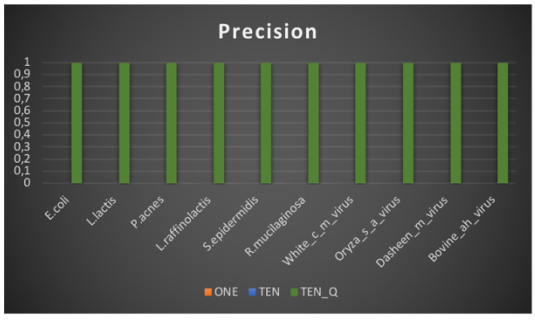
Kraken and Pathseq performances (precision and recall) are displayed in Figure 17. Note that for PathSeq, only values of the third trial (ten percent proportion good quality calls) are shown.

Kraken





PathSeq



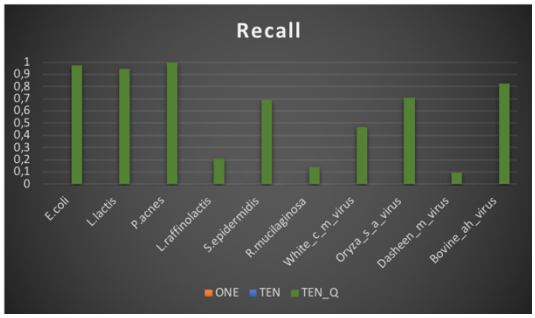


Figure 17: precision and recall values of kraken (left side) and Pathseq (right side).

MetaPhlAn and DRAC performances (precision and recall) are displayed in Figure 18. Precision values were almost unbeatable across the four tools, whereas recall scores fluctuated depending on the species and tool, helping us to rank the tools.

MetaPhlAn DRAC

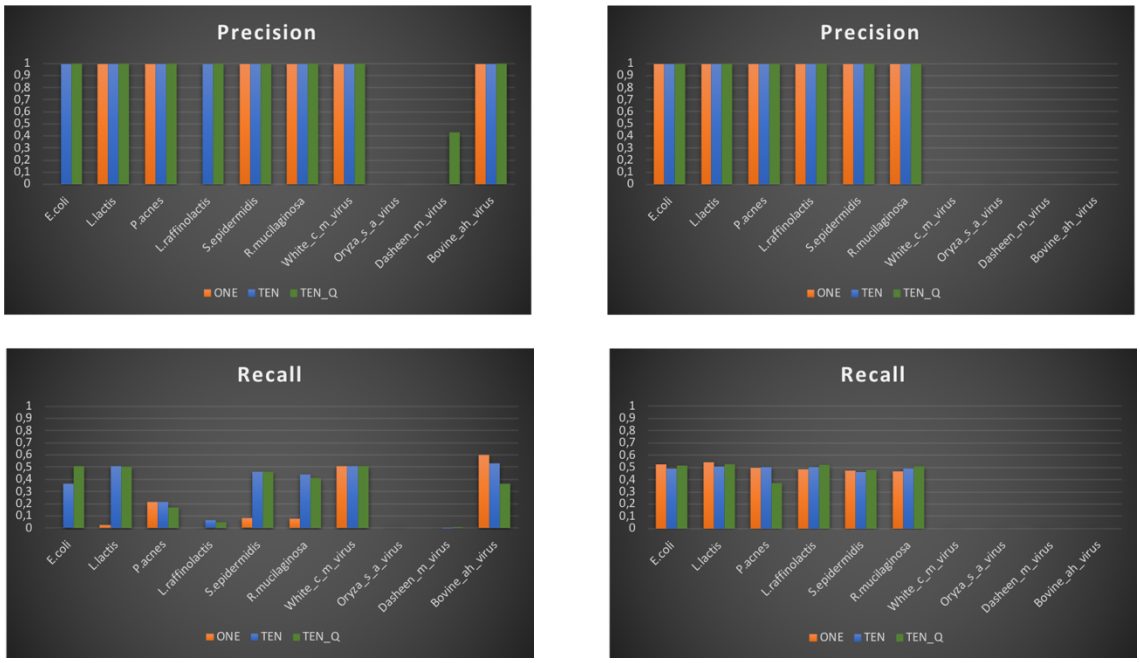


Figure 18: precision and recall values of MetaPhlAn (left side) and DRAC (right side).

Although we computed specificity and negative predicted values score as well, they were both, for the 4 tools almost 1.

Table 8 shows precision and recall values for all the species and values computed only for bacteria species according to the four tools.

Tool	**Recall**	**Recall** (only bacteria)	**Precision**	**Precision** (only bacteria)
Kraken2	0.699	0.762	1	1
PathSeq	0.606	0.659	0.999	0.999
MetaPhlAn	0.297	0.350	0.842	1
DRAC	0.291	0.485	0.600	1

Table 8: Mean recall and precision values from 10% concentration, high quality trial.

6. DISCUSSION

In this master thesis, we performed a benchmarking of three state-of-the-art tools for microorganism detection from RNA samples. We included in this assessment an in-house pipeline (DRAC) built with the same objective. We registered the output results, but also checked the resources needed for each tool. The microbiome profiles were used for a downstream analysis including sample by sample comparison, tumour *vs* solid-adjacent comparison, PCAs, and feature association search using GLM. We also created a set of simulation files (named gold standard) with the same characteristics than the TCGA BAM files using a set of known genomes abundances in order to measure each tool performance.

Our initial intention was to include other tools to the benchmarking, such as STAT, developed by NCBI within the package `sra-tool`. Despite the availability of the tool, it was impossible to set up without publication nor manual. Pandora was another candidate tool, developed by Sakellarios Zairis of the Department of Systems Biology, Columbia University. The short length of TCGA's sequences is the reason why we ended up not using the tool. Pandora relies on an assembly algorithm, which requires longer input reads.

Regarding the origin of data I have been working with, gene expression files, they were coming from a low-biomass set of samples, treated and prepared only for human mRNA capturing what leads to a low reliability for microorganism's RNA/DNA detection. The TCGA mRNA enrichment was performed with polyA capture, but probably this strategy removes a lot of the existing bacterial reads that could been have sequenced using a ribosome depleting method. Also, I would like to highlight that we got rid of all reads classified as human and focused on those that could not be mapped against the human genome. It might happen that human a read could not be correctly mapped, and remained within our input files, leading into a false positive. Therefore, results coming from this investigation and anyone similar, should be considered cautiously.

Regarding our first hypothesis ("Each tool yields different results and similarities will be found on microbiome's abundances, richness and diversity"), the first part of the hypothesis is confirmed. Each tool relies on different structures for presenting their results since they were constructed by different teams according to each developer's needs.

As for the species identified by each tool, only 7 species were common, this low concordance might be due to the specific referral database each tool consults. Note that DRAC was set to work just with bacterial sequences and this fact reduced the number of species and limited the shared species to bacterial ones. Regarding shared species, *P. acnes, E. coli, and S. epidermidis* were among the top identified by all the tools. Their presence fits with what was expected and framed in hypothesis number 5 ("We expect some contaminants") as these 3 species are well-known contaminants.

The concordance among tools is maintained when considering abundance and richness. They all four followed a similar trend. Files with high abundance or richness according to Kraken, had also high values for the rest of the tools, and *vice versa.* Whereas when considering biodiversity, there was not a shared trend. PathSeq and MetaPhlAn values of α-diversity remained constant when sorting files by Kraken biodiversity values. Thus, we can confirm that there was concordance among the tools on richness and abundance but concordance was not high for biodiversity.

Instead of doing the comparison file by file, we could also examine the number of reads for a given bacteria across all files. The example of *E. coli* shown in Figure 4 was a clear case of agreement between PathSeq and Kraken results. This fact endorses the presence of *E. coli*, making this bacterial species more reliable.

The way Kraken's algorithm works makes this tool to be the fastest. Kraken was the tool that took the shortest time to finish, due to the 'exact alignment' technique (Wood, 2014). It saved time by splitting into *k*-mers the query sequence and avoiding the time-consuming process of extending the alignment. It is capable of doing so since its own database is constructed as well with this *k*-mer structure. This fact confirms our second hypothesis.

No clear difference was found when inspecting tumour *vs* solid adjacent samples, even though top species were almost concordant. The standard deviation values from both graphs did widely overlap, reassuring that there was no significant difference. There were two files/samples that contained remarkable high levels of *L. lactis* reads, 37,207 and 54,005, belonging to adjacent solid tissue and tumour files group, respectively. The spike found among normal tissue for *L. lactis* is due to this single file. None of the two cancer associated species *F. nucleatum* and *A. europaeus* were present in the top 20 abundance species. They were not even detected by all tools, being Kraken and PathSeq the only ones to identify them in a really low proportion.

The Principal Component Analysis we performed was not useful to shed light on the microbiome pattern. The variance explained in Kraken and MetaPhlAn were too low and mainly due to extreme values. Only PathSeq's PCA showed a decent variance explained by the first and second components. Two essential conclusions could be extracted from the biplot: 1) the variance explained was greater than the previously seen with the PCA, a fact that was presumably because we only used the top 45 species, excluding the rest. Within the rest of species not considered in the PCA, there were a bunch of "noisy" ones. The second conclusion was that basically three bacteria genera were enough to describe the distribution. *Propionibacterium* explains the horizontal component whereas the vertical was explained by *Escherichia* and *Shigella*.

All across the three PCA plots, we detected outlier samples. Not surprisingly, they happened to be the same samples in all PCAs. The microorganisms distribution of these four samples, were similarly separated from the main clustering

regardless the tool. We decided to present PCA results instead of other methods of 2D/3D representation such as MDS or PCoA because the observed clustering was similar.

Unfortunately, none of the microorganisms studied were significantly associated with gender nor with tumour stage. Although many *p*-values were below the threshold 0.05, the *p*-values adjusted for false discovery rate never met that criterium. Moreover, none of the tools agreed to pinpoint the same microorganism as the one responsible for the association. Each tool yielded a different bunch of species supposedly associated. The most significant one was *Penicillium digitatum*, this fungus would be present in more amount among women than men.

Regarding the simulation results, we saw that Kraken's precision was perfect for all species when quality and abundance were high. Only in this last case, Kraken recognised viruses reads. Surprisingly, Kraken's recall for *E. coli* was very low, a well know model organism whose genome has been intensively explored. Instead of being assigned to *E. coli*, the simulated reads belonging to this species were assigned by Kraken to lowest common ancestor (LCA) genus *Escherichia*. This phenomenon happened to be precisely as a result of over study, as described in Nasko DJ et al, 2018. The increasing size of databases such as NCBI RefSeq has also resulted in more misclassified reads at species level with reliable classifications being pushed higher up the taxonomic tree (for instance genus or family). In general, there was very slight improvement if any when increasing the reads concentration and the reads quality, besides the virus results improvement mentioned before.

PathSeq yielded unbeatable precision scores, and it did not suffer from the same phenomena when classifying *E. coli.* PathSeq yielded a low recall values (both below 0.2) for *L. raffinolactis* and *R. mucilaginosa*, both among the top 4 species initially identified in the TCGA BAM files. On the other hand, and not surprisingly, *Dasheen mosaic virus* obtained a recall bellow 0.2 which is a low value.

MetaPhlAn was, by far, the most benefited of the read's abundance increase. Recall scores clearly got better, with qualitative change for the two newly recognised species (*E. coli* and *L. raffinolactis*). But also, a quantitative increase of the recall score for species already detected. All recall values are either better or equal. The precision improves till 1 for two extra species mentioned before.

MetaPhlAn was the least sensitive tool when comparing bacteria species. This fact confirmed the third hypothesis. When taking into account viruses as well, DRAC was the least sensitive tool. Note that DRAC was not prepared to recognise viruses or fungi. DRAC's recall was quite robust across all bacteria species. It remained constant, close to 0.5 for the 6 species and regardless the concentration and quality. Precision stood up to 1 for the 6 bacterial species as well.

Finally, and summing up, after all we have seen I would recommend Kraken as the best tool in terms of speed and reliability.

7. CONCLUSION

1. There was a low concordance across the tool's microbiome profiles, despite the general agreement in the sample by sample analysis.

2. Kraken was the fastest tool.

3. Kraken had the best performance in terms of recall.

4. MetaPhlAn was the least sensitive tool.

5. No clear difference was found between tumour and peritumoral samples when analysing *F. nucleatum* and *A. europaeus*.

6. No clear association was found so far with gender and tumour stage.

8. FUTURE PLANS

- To improve the simulation/Gold Standard procedure. To generate a database for each species with human and this unique species reads to avoid misclassification due to homologous reads.

- To complete the two first trials (1% low quality base called and 10% low quality base called) of PathSeq in the Gold Standard.

- To reset PathSeq after what we learn with the simulation/Gold Standard study and rerun it on 433 bam files from TCGA, to make sure we leave nothing behind.

- To keep seeking for a possible association with the not explored 77 features of metadata.

- To upload DRAC with viruses databases and rerun the pipeline.

9. FIGURE AND TABLE INDEX

10. GLOSSARY OF ABBREVIATIONS

Abbreviations used during the thesis:

BC	Bladder cancer
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
HPV	Human Papilloma Virus
NGS	Next Generation Sequencing
OTU	Operational Taxonomic Unit
LCA	Lowest Common Ancestor
TCGA	The Cancer Genome Atlas
OR	Odds Ratio

11. BIBLIOGRAPHY

Bi, H. (2019). Urinary microbiota- a potential biomarker and theurapeutic target for bladder cancer. *Journal of Medical Microbiology, 10*, 1471-1478.

Bray F. J. (2018). Global Cancer Statistics . *GLOBOCAN*.

Bucevic, V. (2018). The Urinary microbiome associated with bladder cancer. *Scientific Reports*(8), 12157.

Dematei, A. F. (2017). Angiogenesis in Schistosoma haematobium-associated urinary bladder cancer. *John Wiley & Sons Ltd.*(125(12)), 1056-1062.

Federhen, S. (2012). The NCBI Taxonomy database. *Nucleic Acids Research* , D136–D143.

Gagnaire, A. (2017). Collateral damage: insights into bacterial mechanisms that predispose host cells to cancer. *Nat. Rev. Microbiol*(15), 109-128.
Garrett, W. S. (2015). Cancer and the microbiota. *Science*(348), 80–86.
Gihawi, A. (2019). SEPATH: benchmarking the search for pathogens in human tissue whole genome sequence data leads to template pipeline. *Genome Biology*.
Goodrich JK, D. R. (2014). Conducting a microbiome study . *cell*, 17;158(2):250-262.
JC, W. (2010). A primer on Metagenomics. *PLOS computational biology* .
Metagenomics, A. p. (2010). Wooley JC et al. *PLOS computational Biology* .
Nasko DJ, K. S. (2018). Refseq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. . *Genome Biol*, 19(1).
Nicola Segata, L. W. (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods, 9*, 811–814.
Nurk S, M. D. (2017). metaSPAdes: a new versatile metagenomic assembler. *Genome Research* , 824–834.
Park, S.-J. (2019). A systematic sequencing-based approach for microbial contaminant detection and functional inference. *BMC Biology* .
Robles C, V. R.-C. (2013). Bladder cancer and seroreactivity to BK, JC and Merkel cell polyomaviruses: the Spanish bladder cancer study. *Int J Cancer*.
Selitsky, S. R. (2020). virus expression detection reveals RNA-sequencing contamination in TCGA. *BMC Genomics*.
Soussov, V. S. (2020). NCBI Taxonomy Browser . *https://www.ncbi.nlm.nih.gov/Taxonomy/TaxIdentifier/tax_identifier.cgi*.
Walker, M. (2018). GATK PathSeq: a customizable computational tool for the discovery and identification of microbial sequences in libraries from eukaryotic hosts. *Bioinformatics*(24), 4287-4289.
Wood, D. E. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignment. *Genome Biology, 15*(R46).

Appendix I

gold_standard_PRO_copia.py

```
#!/usr/bin/env python
# coding: utf-8

# In[35]:


import pysam as ps
import numpy as np
import sys
import random


# This script requires:
#
# n:    number of sequences desired for each genome \n
#
# l:    reads length desired \n
#
# q:    guality per read \n
#
# paired:    logical \n
#
# format:    SAM,FASTA_i,FASTA_s \n
#
# genoma_name(s):    genomas des cargados, deben estar en fasta format\n



# In[ ]:
numero=int(sys.argv[1])
length=int(sys.argv[2])
qual=sys.argv[3]
paired=sys.argv[4]
formatt=sys.argv[5]
genomes=sys.argv[6:]


#### Sequences are paired-end. Which means that "mate" is complementary reverse to "
read" chain in 50% of the cases

# In[39]:


def reverscomp(a):
    tr=[]
    for i in a:
        if i=='A':
            tr.append('T')
        if i=='T':
            tr.append('A')
        if i=='C':
            tr.append('G')
        if i=='G':
            tr.append('C')
    return ''.join(tr)[::-1]


# In[64]:


if formatt=='FASTA_i':
    fe=open('generated.fq','wb')
    for genome in genomes: #itinerate genome by genome
        file=open(genome,'r') #cload genome
        f=file.readlines()   #remove header

        header=f[0]
        fi=f[1:]
        fi=''.join(fi)
        fi=fi.replace('\n','')

        for y in range(numero):  #As many times as reads desired for each genome.
```

gold_standard_PRO_copia.py

```
            pi=random.randint(0,len(fi)-((l*2)+120))
                #Randomly select the starting point of the substraction. Taking into
 account the shortest genome plus the two 48nt-sections plus the gap
            gz=random.randint(80,120)
                #Also select randomly the gap size. From 80 to 120 nt long

                ### Extract the sequences themself
            first=fi[pi:pi+l]
            if paired=='TRUE':
                second=fi[pi+gz+l:pi+gz+l*2]
                second_rc=reverscomp(second)   #second_rc is the complementary rever
se. Whereas second is in the same direction and chain than first
                L=[second,second_rc] #Randomly select which one will be printed (50%
)
                second_f=L[int(np.random.randint(2,size=1))]

                a=header[:-2]+'/1'+'\n'+first+'\n'   #first line, corrresponds to fo
rward read
                b=header[:-2]+'/2'+'\n'+second_f+'\n' #line number five, correspond
to backwards mate
                c=l*q+'\n' #second line,quality as long as the read
                d='+'+'\n' #third line, plus sing

                a=a.encode('utf-8')
                b=b.encode('utf-8')
                d=d.encode('utf-8')
                c=c.encode('utf-8')

                #Write the fasta file previously generated.
                fe.write(a)
                fe.write(d)
                fe.write(c)

                fe.write(b)
                fe.write(d)
                fe.write(c)

                fe.write(c)
                del (a,b,c)
            else:  #No paired. Only will print forward reads
                a=header[:-2]+'\n'+first+'\n'
                a=a.encode('utf-8')
                fe.write(a)
                del (a)
    fe.close()


# In[65]:


if formatt=='FASTA_s':  #It will generate two files.
    fe=open('generated_1.fq','wb')
    fe2=open('generated_2.fq','wb')
    for genome in genomes:
        file=open(genome,'r')
        f=file.readlines()

        header=f[0]
        fi=f[1:]
        fi=''.join(fi)
        fi=fi.replace('\n','')

        for y in range(numero):

            pi=random.randint(0,len(fi)-((l*2)+120))
            gz=random.randint(80,120)

            first=fi[pi:pi+l]
            second=fi[pi+gz+l:pi+gz+l*2]
            second_rc=reverscomp(second)
            L=[second,second_rc]
            second_f=L[int(np.random.randint(2,size=1))]
```

gold_standard_PRO_copia.py

```
            a=header[:-2]+'/1'+'\n'+first+'\n'
            b=header[:-2]+'/2'+'\n'+second_f+'\n'
            c=l*q+'\n'
            d='+'+'\n'

            c=c.encode('utf-8')
            d=d.encode('utf-8')
            a=a.encode('utf-8')
            b=b.encode('utf-8')

                #It will send separately to each file,reads and mates respectively.
            fe.write(a)
            fe.write(d)
            fe.write(c)

            fe2.write(b)
            fe2.write(d)
            fe2.write(c)
            del (a,b,c,d)
    fe.close()
    fe2.close()


# In[69]:


if formatt=='SAM':
    fe=open('generated.sam','wb') #We generate a sam file. this time the inside stru
cture is diferent obviously

    for genome in genomes:

        file=open(genome,'r')
        f=file.readlines()

        header=f[0]
        fi=f[1:]
        fi=''.join(fi)
        fi=fi.replace('\n','')

        for y in range(numero):

            pi=random.randint(0,len(fi)-((l*2)+120))
            gz=random.randint(80,120)

            first=fi[pi:pi+l]
            if paired=='TRUE':
                second=fi[pi+gz+l:pi+gz+l*2]
                second_rc=reverscomp(second)
                L=[second,second_rc]

                #we preserve the 11 fields structure.
                a='CNIO:{}:{}'.format(header[1:-1],y)+"\t77\t*\t0\t0\t*\t*\t0\t0\t"+
first+'\t'+l*q+'\n'
                b='CNIO:{}:{}'.format(header[1:-1],y)+"\t141\t*\t0\t0\t*\t*\t0\t0\t"
+L[int(np.random.randint(2,size=1))]+'\t'+l*q+'\n'

                #byte format before writing
                a=a.encode('utf-8')
                b=b.encode('utf-8')

                fe.write(a)
                fe.write(b)
                del (a,b)
            else:
                a='CNIO:{}:{}'.format(header[1:-1],y)+"\t4\t*\t0\t0\t*\t*\t0\t0\t"+f
irst+'\t'+l*q+'\n'
                a=a.encode('utf-8')
                fe.write(a)
                del(a)
    fe.close()
```

Appendix II

tabla_former_kraken.sh

```
#!/bin/bash

#Script former of Kraken's output
#The aim of this for loop is getting as head of the column of reads numbers,the name
 of the file they are coming from.

for FILE in `ls *.krk`;do nombre=$(echo $FILE|cut -c 1-36);echo $nombre>../nombre.tm
p;cut -f3 $FILE>../fila.tmp;cat ../nombre.tmp ../fila.tmp>../$nombre.tmmp;done


cut -f1,4 results_bacteria/cde5a288-ecf4-4894-a3f0-f986568e6833.krk>temp.tmp #Since
all file have the same structure, you could use any of them as template.

echo tax_id    species_name>hola.tmp

cat hola.tmp temp.tmp>template.tmp

#We compile all of the previously generated files.
for FILE in `ls *.tmmp`;do paste -d "\t" template.tmp $FILE>tmp.tmp;mv tmp.tmp templ
ate.tmp;done

mv template.tmp table_absolutefr.tsv #Rename the final document

rm *tmp *tmmp #Clean up before finishing
```

tabla_former_metaphian.py

```
#!/usr/bin/env python
# coding: utf-8

# This script is terminal friendly, as input it takes all the metaphlan output files
# it will merge all of them and then will generate 4 files:
#coverage, average, estimates reads and relative abundance


import pandas as pd
import sys


# In[4]:


#Define the header. The original one started with '#'
header=['clade_name','relative_abundance', 'coverage', 'average_genome_length_in_the
_clade','estimated_number_of_reads_from_the_clade']



#Load output files. U will be the seed up on we built the table
U=pd.read_csv(sys.argv[1],sep='\t',header=[1])
if U.shape==(0,2):   #this is the possible case in wich the tool found nothing and t
he profile has this shape
    U=pd.read_csv(sys.argv[1],sep='\t')
    U['default']=int(0)   #these three columns are to meet the standard number of co
lumns
    U['default1']=int(0)
    U['default2']=int(0)
    U.columns=header
    U.loc[0]=['Unclassified',0,0,0,0]   #One emtpy row for the 5 columns
else:
    U=U.drop(U.index[-1],axis=0)
    U.columns=header                     #replace the header for one easy to work wi
th
U=U[U.clade_name.str.contains('s__')] #we just keep species

name=sys.argv[1][:-12]

#Let subdivide, since we finally would like to obtain 4 several files
estimated=pd.concat([U.clade_name,U.estimated_number_of_reads_from_the_clade],axis=1
)

#we can name each column with the file name
estimated.rename(columns={'estimated_number_of_reads_from_the_clade':name}, inplace=
True)

relative=pd.concat([U.clade_name,U.relative_abundance],axis=1)
relative.rename(columns={'relative_abundance':name}, inplace=True)

coverage=pd.concat([U.clade_name,U.coverage],axis=1)
coverage.rename(columns={'coverage':name}, inplace=True)

average=pd.concat([U.clade_name,U.average_genome_length_in_the_clade],axis=1)
average.rename(columns={'average_genome_length_in_the_clade':name}, inplace=True)


# In[146]:


#M will be every new file iterated at the time, summed up to U.
for i in sys.argv[2:]:

    name=i[:-12]
    #Basicaly repeat the previous steps for all new files, one at the time
    M=pd.read_csv(i,sep='\t',header=[1])

    #By doing this we avoid empty files to go in the final table
    if M.shape==(0,2):
        M=pd.read_csv(i,sep='\t')
        M['default']=int(0)
        M['default1']=int(0)
```

tabla_former_metaphian.py

```
        M['default2']=int(0)
        M.columns=header
        M.loc[0]=['Unclassified',0,0,0,0]
    else:
        M.columns=header
        M=M.drop(M.index[-1],axis=0)
    M=M[M.clade_name.str.contains('s__')]

    m_estimated=pd.concat([M.clade_name,M.estimated_number_of_reads_from_the_clade],
axis=1)
    m_estimated.rename(columns={'estimated_number_of_reads_from_the_clade':name}, in
place=True)


    m_relative=pd.concat([M.clade_name,M.relative_abundance],axis=1)
    m_relative.rename(columns={'relative_abundance':name}, inplace=True)


    m_coverage=pd.concat([M.clade_name,M.coverage],axis=1)
    m_coverage.rename(columns={'coverage':name}, inplace=True)


    m_average=pd.concat([M.clade_name,M.average_genome_length_in_the_clade],axis=1)
    m_average.rename(columns={'average_genome_length_in_the_clade':name}, inplace=Tr
ue)

    acum_estimated=pd.merge(estimated,m_estimated,on='clade_name',how='outer')
    acum_relative=pd.merge(relative,m_relative,on='clade_name',how='outer')
    acum_coverage=pd.merge(coverage,m_coverage,on='clade_name',how='outer')
    acum_average=pd.merge(average,m_average,on='clade_name',how='outer')


    estimated=acum_estimated #we save the infromation accumulated in the seed
    relative=acum_relative
    coverage=acum_coverage
    average=acum_average

    del acum_estimated
    del acum_relative
    del acum_coverage
    del acum_average



a=list(estimated.columns)
a.remove('0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
a.insert(1,'0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
estimated=estimated.reindex(columns=a)
del a

a=list(relative.columns)
a.remove('0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
a.insert(1,'0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
relative=relative.reindex(columns=a)
del a

a=list(coverage.columns)
a.remove('0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
a.insert(1,'0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
coverage=coverage.reindex(columns=a)
del a

a=list(average.columns)
a.remove('0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
a.insert(1,'0082bb8f-6aa4-441c-8c53-43c2b1a34a52')
average=average.reindex(columns=a)
del a
# In[119]:


estimated.fillna(int(0)).to_csv('./estimatedreads_metaphlan.tsv',sep="\t") # fillna
substitute NAN for the specified value.
relative.fillna(0).to_csv('./relativeabundance_metaphlan.tsv',sep="\t")
```

tabla_former_metaphian.py

```
coverage.fillna(0).to_csv('./coverage_metaphlan.tsv',sep="\t")
average.fillna(int(0)).to_csv('./average_metaphlan.tsv',sep="\t")
```

tabla_former_PathSeq.py

```
#!/usr/bin/env python
# coding: utf-8

# In[52]:


import pandas as pd
import numpy as np
import sys #terminal friendly python script


# In[53]:


seed=pd.read_csv('pathseq_microbe_list.txt',sep='\t')
seed=seed.drop('name',axis=1)
seed.columns=['tax_id']   #There will be species withou name since not all species d
iscovered are previously in 'pathseq_microbe_list.txt'
                            #we will work just with tax ids


# In[56]:


for file in sys.argv[1:]:  #for each one of the gatk_output.txt
    tata=pd.read_csv(file,sep='\t')
    tata=tata[tata['type']=='species']  #Filter at species level
    tata= tata[['tax_id', 'reads']].copy()  #substract the two columns of interest f
rom dataframe
    tata.columns=['tax_id',file[:36]]

    tmp=pd.merge(tata,seed,on='tax_id',how='outer')
    tmp=tmp.replace(np.nan,0)  #nan are all of the tax_id from seed,since it has not
 reads column. Plus extra new tax ids
                            #that might show up with each new file
    seed=tmp


# In[63]:


tmp.to_csv('./Gatk_final_table.csv',sep='\t',index=False)
```